

**MAKALAH**  
**STRUKTUR DATA**  
**“Double Linked List”**



Pengampu:  
Angga Gumilang, S.Pd, M.Eng

Oleh:  
Kelompok 6 Golongan C  
E41210985 Indra Bagus Syah Putra  
E41211041 Muhammad Rhomaedi  
E41211194 Lutfi Ameliya Putri

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI JEMBER**  
**2022**

## **KATA PENGANTAR**

Dengan mengucapkan puja dan puji syukur kehadiran Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga makalah STRUKTUR DATA (Double Linked List Circular) ini telah kami selesaikan tepat pada waktunya.

Makalah Double Linked List Circular dibuat sebagai bahan materi Struktur Data. Makalah ini memberikan sedikit gambaran tentang program link list, sehingga makalah ini dapat dijadikan sebagai referensi untuk kita.

Kami sadar bahwa buku ini masih jauh dari sempurna sehingga saran dan kritik pembaca yang sifatnya membangun sangat kami harapkan demi penyempurnaan makalah ini. Terima kasih kami ucapkan kepada teman sekelompok kami yang telah membantu dalam pengerjaan makalah ini.

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>1</b>
<b>BAB I.....</b>	<b>3</b>
<b>PEMBAHASAN .....</b>	<b>3</b>
<b>1.1 Pengertian .....</b>	<b>3</b>
• Double Linked List Circular .....	3
• Double Linked List Non-Circular .....	3
<b>1.2 Representasi Linked List .....</b>	<b>4</b>
• Representasi Simpul.....	4
• Deklarasi Global.....	4
<b>1.3 Operasi pada Double Linked List.....</b>	<b>4</b>
• Operasi Sisip .....	4
• Operasi Hapus .....	5
<b>1.4 Contoh Program dan Output .....</b>	<b>6</b>
• Source Code .....	6
• Ouput.....	8
<b>BAB II .....</b>	<b>9</b>
<b>KESIMPULAN .....</b>	<b>9</b>
<b>REFERENSI.....</b>	<b>10</b>

## BAB I

### PEMBAHASAN

#### 1.1 Pengertian

Double linked list adalah suatu linked list yang mempunyai 2 penunjuk yaitu penunjuk ke simpul sebelumnya dan simpul berikutnya. Pada implementasinya, terdapat dua variasi double linked list yaitu:

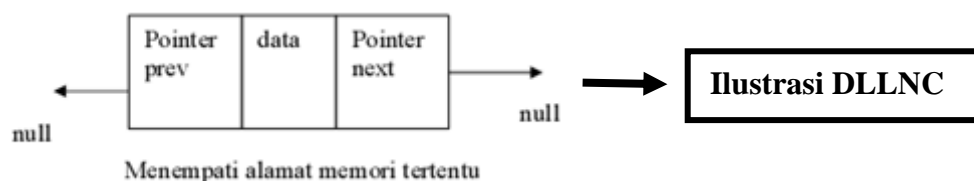
- Double Linked List Circular

- Double : Field Pointernya terdiri dari dua buah dan dua arah, yaitu prev dan next.
- Linked List : Node-node tersebut saling terhubung satu sama lain.
- Circular : Pointer next dan prev-nya menunjuk ke dirinya sendiri.

Double Linked List Circular adalah linked list dengan menggunakan pointer, dimana setiap node memiliki 3 field, yaitu 1 field pointer yang menunjuk pointer berikutnya (next), 1 field menunjuk pointer sebelumnya (prev), serta sebuah field yang berisi data untuk node tersebut. Double Linked List Circular pointer next dan prev nya menunjuk ke dirinya sendiri secara circular.

- Double Linked List Non-Circular

- Double : artinya field pointer-nya dua buah dan dua arah, ke node sebelum dan sesudahnya.
- Linked List : artinya node-node tersebut saling terhubung satu sama lain.
- Non Circular : artinya pointer prev dan next-nya akan menunjuk pada NULL



Setiap node pada linked list mempunyai field yang berisi data dan pointer ke node berikutnya & ke node sebelumnya. Untuk pembentukan node baru, mulanya pointer next dan prev akan menunjuk ke nilai NULL. Selanjutnya pointer prev akan menunjuk ke node sebelumnya, dan pointer next akan menunjuk ke node selanjutnya pada list.

## 1.2 Representasi Linked List

- Representasi Simpul

```
Typedef struct simpul Dnode;
```

```
Struct simpul {
```

```
    Int data;
```

```
    Dnode *next;
```

```
    Dnode *prev;
```

```
};
```

- Deklarasi Global

```
Dnode *head = null;
```

```
Dnode *tail = null;
```

```
Dnode *baru;
```

## 1.3 Operasi pada Double Linked List

- Operasi Sisip

- a. Penyisipan didepan

Operasi ini berguna untuk menambahkan satu simpul baru diposisi pertama. Langkah pertama untuk menambahkan data adalah pembuatan simpul baru dan mengisinya dengan data pada field infonya. Pointer yang menunjuk ke simpul tersebut dipanggil dengan nama baru.

- b. Penyisipan ditengah

Operasi penyisipan data ditengah linked list adalah suatu operasi menambah data di posisi tertentu di dalam linked list. Karena double linked list memiliki dua pointer sambungan, maka penyisipan bisa dilakukan sebelum data tertentu atau sesudah data tertentu, berbeda dengan single linked list yang hanya memiliki satu pointer sambungan, yaitu sambungan next.

- c. Penyisipan dibelakang

Operasi ini berguna untuk menambahkan elemen baru diposisi akhir. Langkah pertama untuk penambahan data adalah pembuatan elemen baru dan pengisian nilai infonya. Pointer yang menunjuk ke data tersebut dipanggil dengan nama baru.

- Operasi Hapus

- a. Hapus dimpul awal

Penghapusan data diawal adalah proses penghapusan simpul pertama (yang ditunjuk oleh variabel pointer awal), sehingga variabel pointer awal akan berpindah ke simpul berikutnya. Ada tiga kondisi yang perlu diperhatikan, yaitu kondisi linked list masih kosong, kondisi linked list hanya memiliki satu data, dan linked list yang memiliki data lebih dari satu elemen.

- b. Hapus simpul akhir

Penghapusan data diakhir adalah proses penghapusan pada simpul akhir (yang ditunjuk pointer akhir) dengan cara mengcopy pointer akhir dengan pointer hapus dan memindahkan pointer akhir ke tetangga kirinya dan data pun dapat dihapus.

- c. Hapus simpul tengah

Penghapusan data ditengah merupakan proses penghapusan data dengan cara user memasukkan data/posisi data yang akan dihapus dan ditunjuk oleh pointer hapus. Untuk menghapus data tersebut dibutuhkan sebuah pointer bantu yang menunjuk data sebelum data yang akan dihapus. Dengan tujuan sebagai penyambung kembali data sebelum dan sesudah data yang akan dihapus.

## 1.4 Contoh Program dan Output

- **Source Code**

```
package struktur_data;

public class DoubleLinkedList {
    Node head;

    class Node{
        int data;
        Node prev;
        Node next;
        Node(int d) {
            data = d;
        }
    }

    public void prepend(int new_data) {
        Node new_node = new Node(new_data);
        new_node.next = head;
        new_node.prev = null;
        if (head != null)
            head.prev = new_node;
        head = new_node;
    }

    public void append(int new_data){
        Node new_node = new Node(new_data);
        Node last = head;
        new_node.next = null;
        if (head == null) {
            new_node.prev = null;
            head = new_node;
            return;
        }
        while(last.next != null)
            last = last.next;
        last.next = new_node;
    }
}
```

```

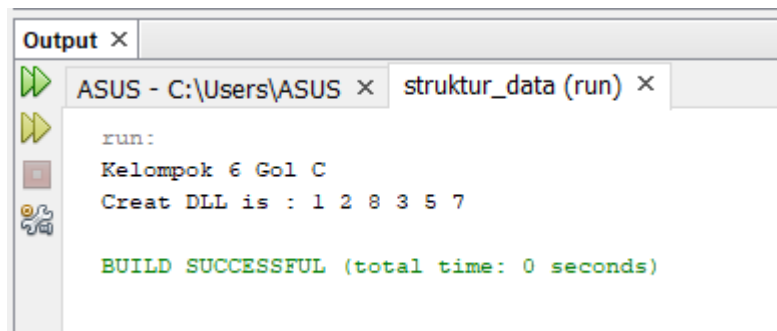
        new_node.prev = last;
    }
    public void insertafter(Node prev_node,int new_data) {
    if(prev_node == null) {
        return ;
    }
    Node new_node = new Node(new_data);
    new_node.next = prev_node.next;
    prev_node.next = new_node;
    new_node.prev = prev_node;
    if (new_node.next != null)
        new_node.next.prev = new_node;
    }
    public void printlist(Node node) {
        Node temp = null;
        while(node != null) {
            System.out.print(node.data + " ");
            temp = node;
            node = node.next;
        }
    }
    public static void main(String[] args) {
        DoubleLinkedList dll = new DoubleLinkedList();
        dll.prepend(2);
        dll.append(3);
        dll.append(5);
        dll.append(7);
        dll.prepend(1);
        dll.insertafter(dll.head.next, 8);
        System.out.println("Kelompok 6 Gol C");
        System.out.print("Creat DLL is : ");
        dll.printlist(dll.head);
        System.out.println("\n");
    }

```



}

- **Ouput**



## **BAB II**

### **KESIMPULAN**

Double Link List adalah elemen-elemen yang dihubungkan dengan dua pointer dalam satu elemen dan list dapat melintas baik di depan atau belakang.

## REFERENSI

- D.Anissa. 2021. *Makalah Linked List*. <https://idoc.pub/documents/makalah-linked-list-k6nq7xgqe14w>
- Koem21. 2010. *Double Lingked List Non Circular*. <https://id.scribd.com/doc/39883606/Double-Linked-List-Non-Circular>